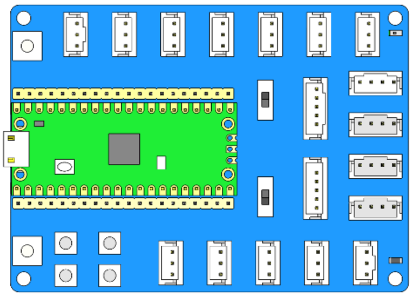
Lesson 4 How to use the Servo

4.1 Components & Parts

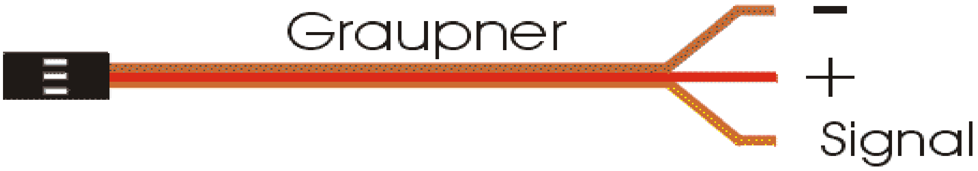
Components	Quantity	Picture	Remark
Raspberry Pi Pico	1		
USB cable	1		
MG90S Servo	1		Not included in the Kit, you can prepared by yourself
Pico Expansion board	1		
Dupont wire	3pin		

4.2 Knowledge of MG90s Servo

4.2.1 Servo Physical map

As shown in the figure below, a single MG90S Micro Servo consists of three parts: servo, servo plate, and screws

4.2.2 Pins



4.2.3 Parameters

4.2.3.1 Electrical parameters

No.	（Item）	5V
4.2.3.1.1	No-load current	120±20mA
4.2.3.1.2	load current	160±20mA
4.2.3.1.3	No-load speed	0.11sec/60º

4.2.3.1.4	Quiescent Current	10mA
4.2.3.1.5	No-load life	50000 times
4.2.3.1.6	stall torque	1.5Kg.-cm
4.2.3.1.7	Stall current	≦1A
4.2.3.1.8	Temperature drift (ambient temperature 25°)	≦1°
4.2.3.1.9	temperature resistance	-20°C

4.2.3.2 Structural parameters

No.	（Item）	Specification
4.2.3.2.1	Physical dimension	32.6*12.1*22.5mm
4.2.3.2.2	Weight	12G
4.2.3.2.3	Mechanical limit angle	360°
4.2.3.2.4	Gear type	Grade 5 Metal Gear Set
4.2.3.2.5	Output tooth spline	20T
4.2.3.2.6	Wire length	250±5mm
4.2.3.2.7	Gear virtual position	≦2°
4.2.3.2.8	Wire	50C/0.08 OD1.0 JR Connector
4.2.3.2.9	Swing arm	One Word Rocker Cross rocker Flower rocker
4.2.3.2.10	Motor	DC motor
4.2.3.2.11	Shell material	PC

4.2.3.3 Control parameter

No.	（Item）	Specification
4.2.3.3.1	Control signal	PWM cycle 50HZ
4.2.3.3.2	Pulse width range	1000us-2000us
4.2.3.3.3	Amplifier type	Number
4.2.3.3.4	Rotation angle	90°±3°(when 1000～2000usec)
4.2.3.3.5	Midpoint	1500usec
4.2.3.3.6	Dead zone width	≦5usec
4.2.3.3.7	Direction of rotation	Clockwise(when 1000～2000usec)
4.2.3.3.8	Return error	≦1°
4.2.3.3.9	Over-operating angle range	180°(500-2500usec)
4.2.3.3.10	Angle error on both sides	≦5°

4.2.4 Working Principle

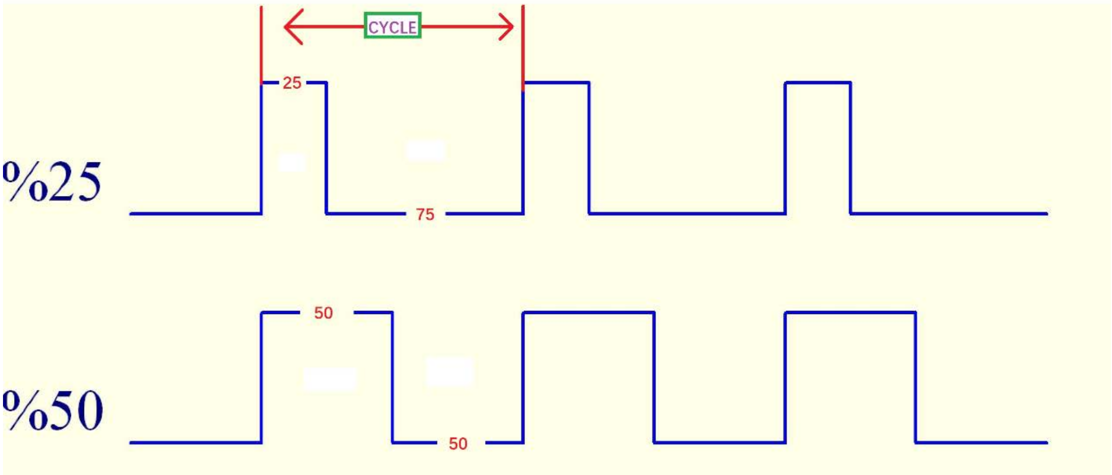
The control signal of the servo is a PWM signal with a period of 20ms, in which the pulse width is from 0.5ms-2.5ms, and the corresponding position of the servo is 0-180 degrees, which changes linearly.Provide it with a certain pulse width, and its output shaft will remain at a corresponding angle, no matter how the external torque changes, until a new pulse signal of different width is provided to it, it will change the output angle to the new corresponding position. There is a reference voltage inside the servo, which generates a reference signal with a period of 20ms and a width of 1.5ms. There is a comparator that compares the applied signal with the reference signal to determine the direction and size, thereby generating the rotation signal of the motor. The internal control circuit board of the servo receives the control signal from the signal line, and controls the rotation of the motor. The motor drives a series of gear sets, which are driven to the output steering wheel after deceleration. The output shaft of the servo is connected to the position feedback potentiometer. When the steering wheel rotates, it drives the position feedback potentiometer. The potentiometer will output a voltage signal to the control circuit board for position feedback. The control circuit board determines the rotation direction and speed of the motor according to the position of the output shaft, so that the output shaft stops when it reaches the target.

4.2.5 PWM Control

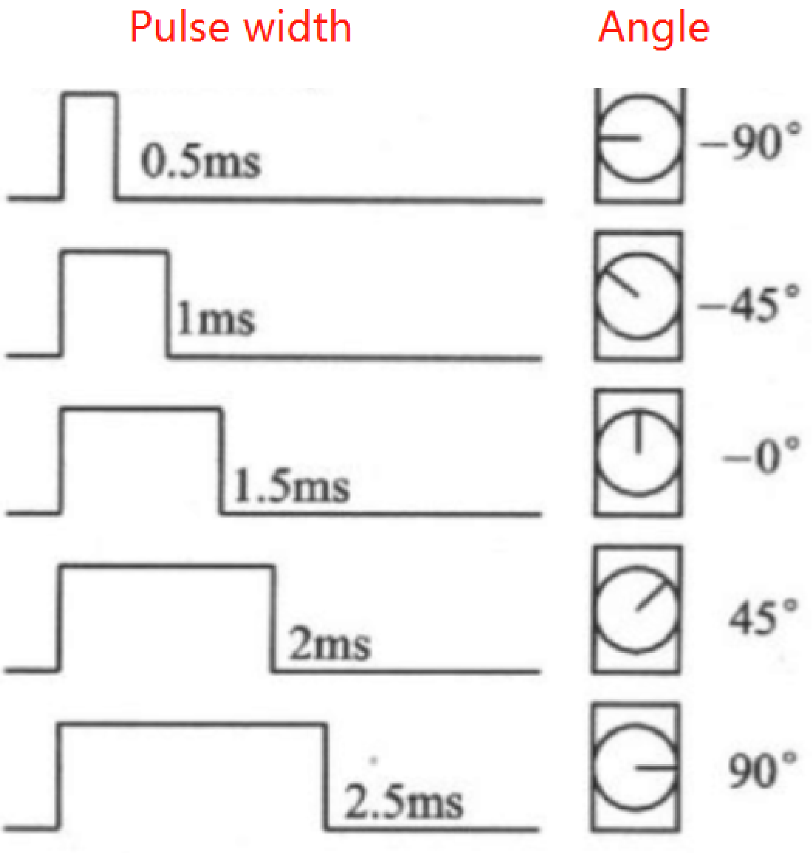
4.2.5.1 PWM Introduction

Pulse width modulation (PWM) refers to the use of the digital output of a microprocessor to control an analog circuit, and is a method of digitally encoding the level of an analog signal.

PWM (Pulse Width Modulation): Pulse Width Modulation
Pulse: square wave, frequency (freq)
Width: the width of the high level, the duty cycle (duty)
Cycle: CYCLE
Duty cycle: the proportion of high level (100&%)
Duty cycle static diagram:



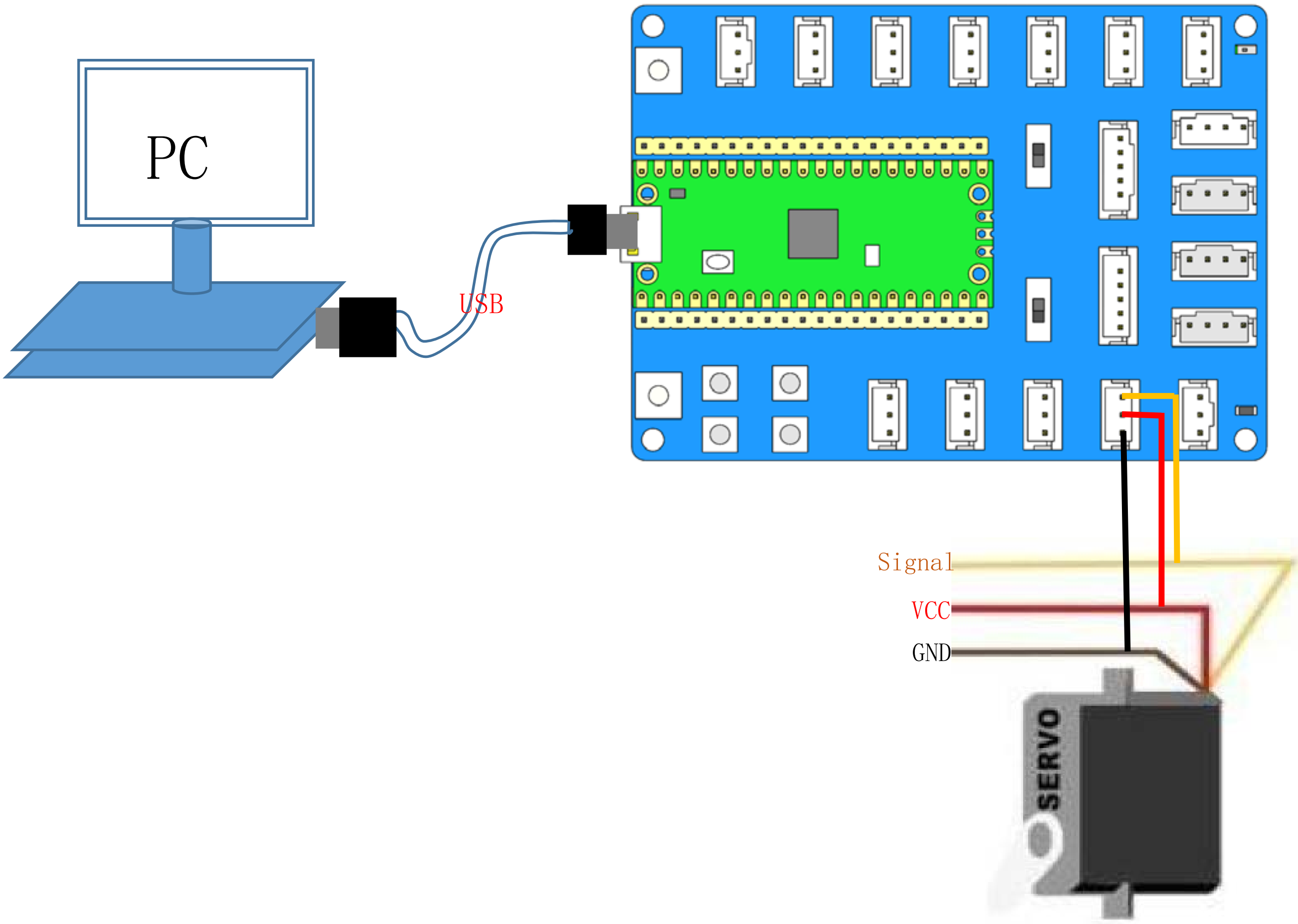
The relationship between the output angle of MG90S Micro Servo and the pulse width of the input signal



CYCLE=20ms
Duty_16=65532*Pulse Width/CYCLE
The relationship between the corresponding pulse width and duty_u16:

Pulse Width(ms)	Duty_16
0.5	1638
1	3276
1.5	4914
2	6552
2.5	8192

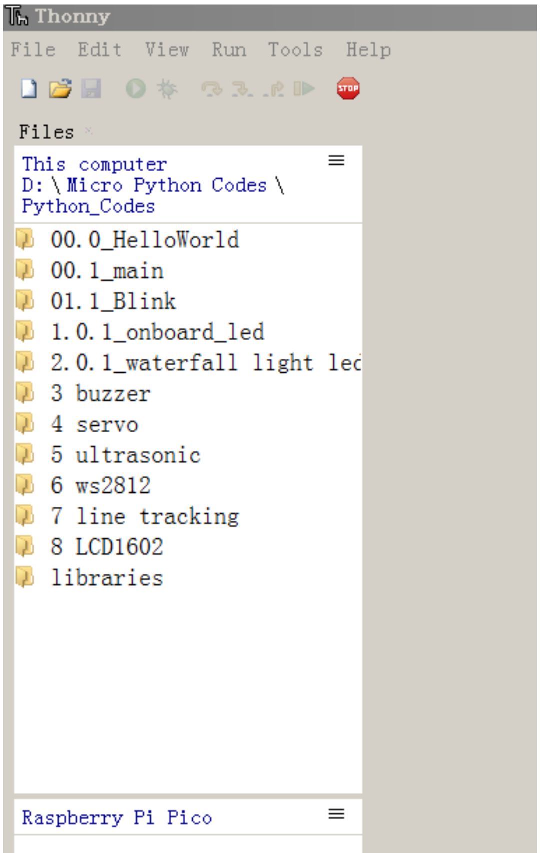
4.3 Circuit



4.4 Run the program

4.4.1 Codes used in this tutorial are saved in "Pico Expansion Kit Tutorial\Lessons for Python\Python_Codes".You can move the codes to any location. For example, we save the codes in Disk(D) with the path of "D:/ Micro Python codes".

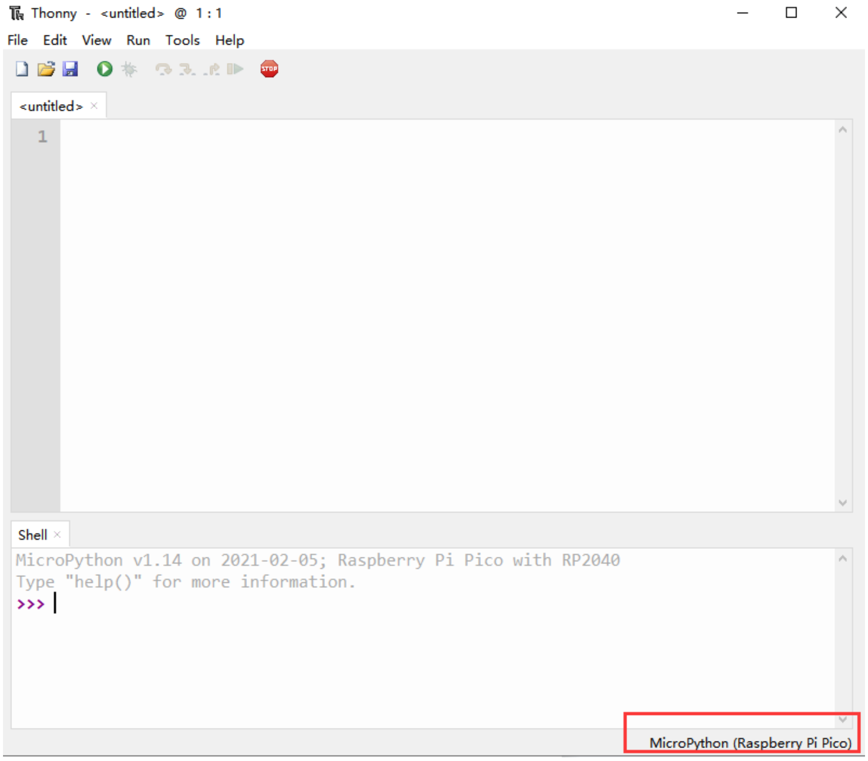
4.4.2 Open "Thonny", click "View"--"Files"--"This computer"--"D:"--"Micro Python Codes"--"Python_Codes".



This computer: The file area of the personal computer.

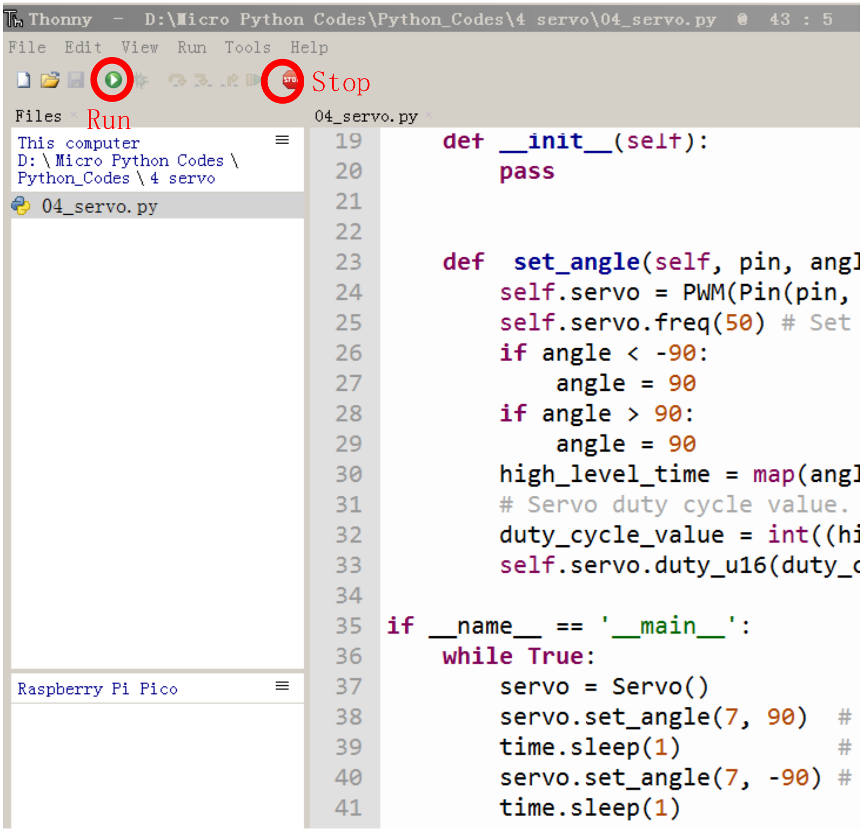
Raspberry pi Pico: Pico file area, the code saved in Pico can be viewed in this area.

4.4.3 Confirm that "MicroPython (Raspberry Pi Pico)" is displayed in the lower right corner. If it is not, please click the font in the lower right corner to select "MicroPython(Raspberry Pi Pico)" mode.


Thonny - <untitled> @ 1 : 1
File Edit View Run Tools Help
<untitled>
1
Shell
MicroPython (Raspberry Pi Pico)

```
MicroPython v1.14 on 2021-02-05; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> 
```

4.4.4 Double-click the code "04_buzzer.py" required for this course. The content of the code will be displayed in the interface on the right.


Thonny - D:\Micro Python Codes\Python_Codes\4 servo\04_servo.py @ 43 : 5
File Edit View Run Tools Help
Run
Stop
Files
04_servo.py
This computer
D:\Micro Python Codes\Python_Codes\4 servo
04_servo.py
Raspberry Pi Pico

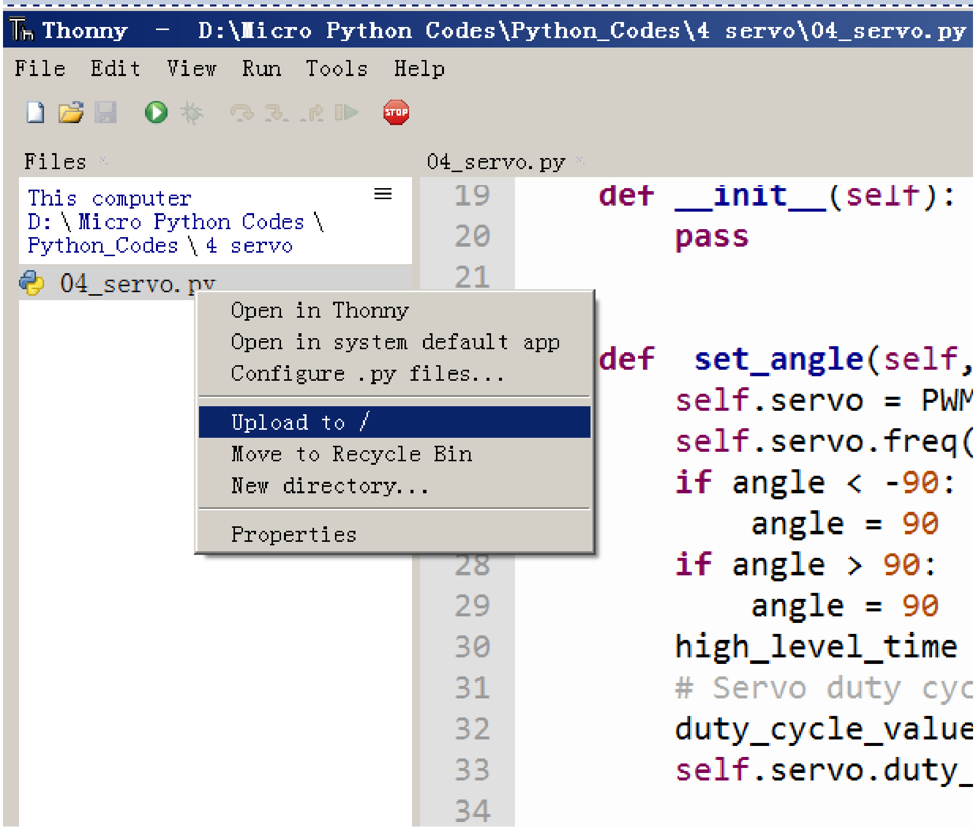
```
    def __init__(self):
        pass


    def  set_angle(self, pin, angl
        self.servo = PWM(Pin(pin,
        self.servo.freq(50) # Set
        if angle < -90:
            angle = 90
        if angle > 90:
            angle = 90
        high_level_time = map(angl
        # Servo duty cycle value.
        duty_cycle_value = int((hi
        self.servo.duty_u16(duty_c

if __name__ == '__main__':
    while True:
        servo = Servo()
        servo.set_angle(7, 90)  #
        time.sleep(1)           #
        servo.set_angle(7, -90) #
        time.sleep(1)
```

4.4.5 Click the Run button to run the program,then the servo in the circuit will working, you can see it rotate 180 degree repeatly

4. 4. 6 Click the stop button to stop the program,the servo will stop rotating.

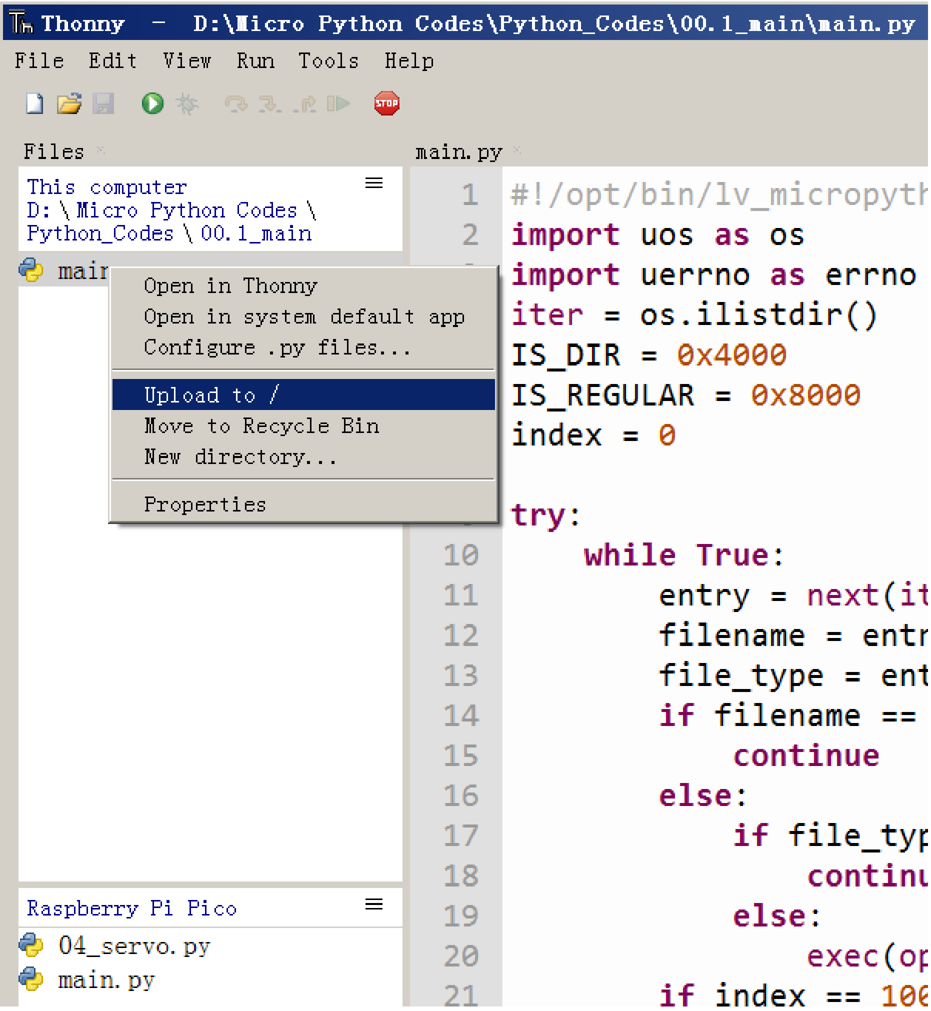
Above is the way to run the program on line, we can also run the program off line

4.4.7 Select the “04_servo.py”,then right click the mouse,click the“Upload to /”,it will upload the “04_servo.py” file to the Pico.


Thonny - D:\Micro Python Codes\Python_Codes\4 servo\04_servo.py
File Edit View Run Tools Help
Files
04_servo.py
This computer
D: \ Micro Python Codes \ Python_Codes \ 4 servo
04_servo.py
Open in Thonny
Open in system default app
Configure .py files...
Upload to /
Move to Recycle Bin
New directory...
Properties

```
    def __init__(self):
        pass


    def  set_angle(self,
        self.servo = PWM
        self.servo.freq(
        if angle < -90:
            angle = 90
        if angle > 90:
            angle = 90
        high_level_time
        # Servo duty cyc
        duty_cycle_value
        self.servo.duty_
```

4.4.5 Expand the folder “00.1_main”,select the “main.py”,then right click the mouse,click the“Upload to /”,it will upload the “main.py” file to the Pico.


Thonny - D:\Micro Python Codes\Python_Codes\00.1_main\main.py
File Edit View Run Tools Help
Files
main.py
This computer
D:\Micro Python Codes\Python_Codes\00.1_main
main
Open in Thonny
Open in system default app
Configure .py files...
Upload to /
Move to Recycle Bin
New directory...
Properties
Raspberry Pi Pico
04_servo.py
main.py

```
#!/opt/bin/lv_micropyth
import uos as os
import uerrno as errno
iter = os.ilistdir()
IS_DIR = 0x4000
IS_REGULAR = 0x8000
index = 0

try:
    while True:
        entry = next(it
        filename = entr
        file_type = ent
        if filename ==
            continue
        else:
            if file_typ
                continu
            else:
                exec(op
        if index == 10
```

Make sure the 04_servo.py and main.py are upload to the pico

Disconnect the USB cable and Thonny, then supply power to the Pico Expansion board(you can connect the pico to 5v adapter via an usb cable, or you can connect the Pico expansion board power port to DC supply via the red-black wire). you can see that the Mg90s servo rotate 180 degrees repeatedly

4.5 Code

```
from machine import Timer,Pin, PWM, ADC
# from machine import time_pulse_us
import time, array
import math

# mapping function
def map(x,in_max, in_min, out_max, out_min):
    return (x - in_min)/(in_max - in_min)*(out_max - out_min) + out_min

'''
PERIOD = ffff = 65535
freq = 50
'''
class Servo():
    pwm_max = 2500
    pwm_min = 500
    period = 65535 # oxFFFF

    def __init__(self):
        pass


    def   set_angle(self, pin, angle):    # Pin:Servo GPIO pins, angle:Servo rotation angle -90~90.
        self.servo = PWM(Pin(pin, Pin.OUT))
        self.servo.freq(50) # Set servo Freq.
        if angle < -90:
            angle = 90
        if angle > 90:
            angle = 90
        high_level_time = map(angle, 90, -90, self.pwm_max, self.pwm_min)
        # Servo duty cycle value.
        duty_cycle_value = int((high_level_time/20000)*self.period)
        self.servo.duty_u16(duty_cycle_value)

if __name__ == '__main__':
    while True:
        servo = Servo()
        servo.set_angle(7, 90)    # Rotate to 90 degrees.
        time.sleep(1)                    # delay 1s.
```

```
servo.set_angle(7, -90) # Rotate to -90 degrees.
time.sleep(1)
```

4.6 What's Next?

THANK YOU for participating in this learning experience!

If you find errors, omissions or you have suggestions and/or questions about this Lesson, please feel free to contact us: cokoino@outlook.com

We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our website. We will continue to launch fun, cost-effective, innovative and exciting products.

http://cokoino.com/

Thank you again for choosing Cokoino products.